The Complexity of Translationally-Invariant Low-Dimensional Spin Lattices in 3D

Johannes Bausch*[1] and Stephen Piddock†[2]

[1]DAMTP, University of Cambridge
[2]School of Mathematics, University of Bristol

September 21, 2018

In this paper, we consider spin systems in three spatial dimensions, and prove that the local Hamiltonian problem for 3D lattices with face-centered cubic unit cells, 4-local translationally-invariant interactions between spin-3/2 particles and open boundary conditions is $\mathsf{QMA}_{\mathsf{EXP}}$-complete. We go beyond a mere embedding of past hard 1D history state constructions, and utilize a classical Wang tiling problem as binary counter in order to translate one cube side length into a binary description for the verifier input. We further make use of a recently-developed computational model especially well-suited for history state constructions, and combine it with a specific circuit encoding shown to be universal for quantum computation. These novel techniques allow us to significantly lower the local spin dimension, surpassing the best translationally-invariant result to date by two orders of magnitude (in the number of degrees of freedom per coupling). This brings our models en par with the best non-translationally-invariant construction.

Introduction and Motivation

Hamiltonian operators are used ubiquitously to describe physical properties of multi-body quantum systems, and are of paramount interest for an array of disciplines ranging from theoretical computer science, to experimental and condensed matter physics. While computer scientists are interested in the computational power of different models (e.g. Hamiltonian quantum computers), for physicists it is important to calculate the structure of the low-energy spectrum of quantum systems. One of the most basic, yet fundamental such question is to estimate the ground state energy of a many-body spin system with low-range interactions, formally known as the *local Hamiltonian problem*.

Kitaev's seminal paper proving quantum-NP-hardness of the local Hamiltonian problem for the case that each interaction couples at most five spins [1] motivated significant progress towards understanding the computational complexity that arises in different variants of the local Hamiltonian problem [2–10]. These results are especially interesting from a computational perspective, answering which families of Hamiltonians are "complicated enough" to perform universal quantum computation [11, 12]. Analysing the energy levels of the resulting hard instances often required the development of novel mathematical techniques, which are of independent interest e.g. in the context of spectral analysis of stochastic processes, or perturbation theory. Yet from the perspective of experimental physics and material sciences, the resulting many-body quantum systems are too contrived to be of relevance; either the local spin dimension is vast, the coupling strengths vary from site to site, or the interaction graphs are not geometrically local.

Moreover, while 1D results are interesting and in a sense the most fundamental models to study (as any 1D hardness result directly implies hardness of the corresponding higher-dimensional constructions), most condensed matter systems are in fact two- or three-dimensional, and the comparison of local dimension between the best non-translationally invariant results in 1D and 2D —8 [7], and 2 [3], respectively—indicates that moving beyond 1D allows a significant reduction of the lattice spins' dimension. It is thus a natural question to ask whether one can go beyond a simple reduction from previously-known 1D results, by exploiting these extra dimensions in a non-trivial way (i.e. beyond a simple embedding), but at the same time retaining nice physical properties such as a regular lattice structure and translational symmetries. We can even go further: is there a family of Hamiltonians on a physically realistic 3D crystal lattice with a QMA-hard ground state? This question is highly relevant, since such crystal structures are found ubiquitously in nature (e.g. face-centered cubic lattices for sodium chloride, or body-centered cubic cesium chloride crystals).

In this paper, we prove that the local Hamiltonian problem remains computationally hard, even for a face-centered cubic lattice of spin-3/2 particles with geometrically 4-local translationally-invariant interactions, and open boundary conditions.

It is clear that there is always a trade-off between local dimension and interaction range: a Hermitian operator coupling k spins of dimension d each has d^{2k} real degrees of freedom. In 1D and for 2-local interactions, the best-known construction to date is [7] with 8-dimensional qudits and nearest-neighbour interactions; for each coupled pair of qudits, one Hermitian operator thus has $8^2 \times 8^2 = 16384$ free real parameters. Enforcing translational invariance, we can regard e.g. [8]—nearest-neighbour interactions between spins of dimension ≈ 50—which would give roughly $(50^2)^2 \approx 6 \times 10^6$ parameters to choose from.

The construction we propose in this paper with at most 4-local interactions between spins of dimension 4 yields 4^8 degrees of freedom, a roughly two orders-of-magnitude improvement over a straightforward embedding of the best one-dimensional construction, and en par with the best non-translationally-invariant result. It also shows that there is only about three orders of magnitude left between this construction and spin systems that we encounter every day (e.g. nearest-neighbour, spin 1).

Main Result

The family of spin systems we study are described by a Hamiltonian on a face-centered cubic (cF) lattice as shown in fig. 1. More precisely, we start with a finite cubic lattice Λ, where each vertex *and each face* carries a 4-dimensional spin $\mathcal{H}_{\text{loc}} = \mathbb{C}^4$; the overall Hilbert space $\mathcal{H}$ is then the tensor product of all spins. For a geometrically local Hamiltonian $\mathbf{h}$ acting on k neighbouring spins (on vertices, faces, or both), we denote with $\mathbf{h}^{\vec{x}}$ the k-local operator $\mathbf{h}$ when offset by a lattice vector $\vec{x} \in \Lambda$, and acting trivially everywhere else; in case that $\mathbf{h}^{\vec{x}}$ protrudes out of Λ, we set $\mathbf{h}^{\vec{x}} \equiv \mathbf{0}$. For a finite index set I, we consider Hamiltonians of the form

$$\mathbf{H} = \sum_{i \in I} \left(c_i \sum_{\vec{x} \in \Lambda} \mathbf{h}_i^{\vec{x}} \right), \tag{1}$$

where each $\mathbf{h}_i^{\vec{x}}$ couples at most 4 spins, either within a single unit cell, or between neighbouring unit cells. By construction, this Hamiltonian is translationally-invariant, and features open boundary conditions since we do not place special interactions at faces, edges or corners of the lattice cuboid.

The index set I does not depend on the size of the lattice, and neither do any of the $\mathbf{h}_i$; we allow the $c_i = c_i(|\Lambda|)$ to depend on the system size $|\Lambda| = W \times H \times D$, but require any $c_i/c_j \in [\Omega(1/\operatorname{poly}|\Lambda|), \mathrm{O}(\operatorname{poly}|\Lambda|)]$. This allows us to define a variant of the local Hamiltonian problem where the input is given by a description of the local terms of a Hamiltonian as in eq. (1) (i.e. the matrix entries of the local terms $c_i \times \mathbf{h}_i$, up to polynomial precision), as well as the three side-lengths W, H and D of the lattice. Moreover, we are given two parameters $\alpha < \beta$ satisfying $\beta - \alpha = \Omega(1/\operatorname{poly}|\Lambda|)$, and a promise that the ground state energy of $\mathbf{H}$ is either smaller than α, or larger than β. The local Hamiltonian problem is then precisely the question of distinguishing between these two cases, and we prove the following main theorem.

Theorem 1. *The local Hamiltonian problem is* $\mathsf{QMA_{EXP}}$*-complete, even for translationally-invariant 4-local interactions on a 3D face-centered cubic spin lattice (fig. 1) with local dimension 4, and open boundary conditions.*

$\mathsf{QMA_{EXP}}$ is similar to QMA, the quantum analogue of NP, but with an exponential-time verifier instead of polynomial-time—a necessary technicality for any translationally invariant result [8, 13], since an n-qudit instance can only encode $\operatorname{poly}(\log n)$ bits of information (in this case the side lengths of the lattice, which encode the input in unary).

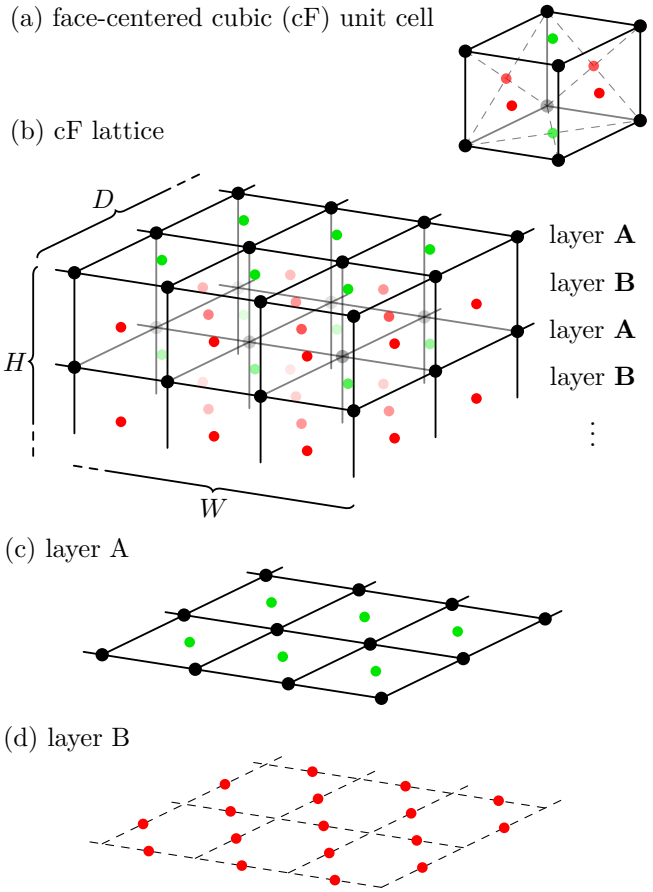


Figure 1: *Face-centered cubic crystal lattice. All vertices and faces carry spin-3/2 particles; the red and green sublattice spins sit on the faces defined by the black lattice.*

In essence, while a $\mathsf{QMA_{EXP}}$-hard problem can be *verified* in exponential time on a quantum computer, just as in the P vs. NP case it is not expected to be *solved* as efficiently (see appendix A for details).

We give a rigorous proof of theorem 1 in appendix G; in the following, we want to give a high-level exposition of the ideas and proof techniques which we employ. As in past hardness results, we present an explicit construction of a family of $\mathsf{QMA_{EXP}}$-hard instances of this variant of the local Hamiltonian problem. We will make use of two types of local terms, tiling and history state Hamiltonians, both of which have been studied extensively, but mostly independently of each other. In our work, we will utilize each method to its strength: the classical tiling terms will be used to encode the bootstrapping mechanism responsible for the large local dimension in prior work, while the history state terms will be used as a means of embedding the quantum computation part. First we will briefly recap these methods.

History State Construction. By definition, a promise problem Π is in $\mathsf{QMA_{EXP}}$ if there exists a BQEXP quantum circuit—called "verifier"—such that for any YES-instance $l \in \Pi$, there exists a poly-sized quantum state—called "witness"—which the verifier accepts with probability $\geq 2/3$; or if l is a NO-instance, all poly-sized witnesses are rejected with high probability. The exact constant used here is not

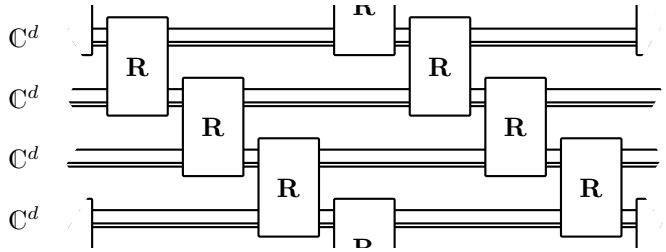


Figure 2: *Circuit diagram of a QRM with a ring of four dimension d qudits. The intuition behind proving universality for QRMs is to encode a classical (reversible) Turing machine's action into the unitary* $\mathbf{R}$*; depending on the internal state—which is stored on classical lanes of the circuit (double lines)—a controlled unitary is applied to the pair of qubits stored on the quantum lanes (single lines). Special flags also stored on the classical lanes indicate where the unitary* $\mathbf{R}$ *acts non-trivially in its next round. In this way, the Turing machine can "write out" and apply a uniform family of quantum circuits in one go. QRMs are thus quantum Turing complete for a uniform complexity class, given that the ring scales sufficiently quickly with the input.*

important, as for any polynomial p, one can always amplify a $\mathsf{QMA}_{\mathsf{EXP}}$ promise problem such that the distinction works with probability $\geq 1 - 1/3^{p(|l|)}$ for an instance $l \in \Pi$ with size $|l|$ (cf. remark 1).

We further know that for any $\mathsf{QMA}_{\mathsf{EXP}}$ promise problem, we can alternatively obtain a so-called *quantum ring machine* (QRM) as verifier (lemma 1). In brief, a QRM is a fixed unitary $\mathbf{R}$ on $(\mathbb{C}^d)^{\otimes 2}$, which acts cyclicly on a ring of n dimension d qudits. Borrowing terminology from Turing machines (TM)—which are used to prove universality of the QRM model—we call the unitary $\mathbf{R}$ the *head* of the QRM, and the qudit ring is essentially a TM tape with cyclic boundary conditions. Fig. 2 depicts such a QRM and its action in circuit notation.

We take a specific 2-qubit quantum gate $\mathbf{G}$ and prove it to be universal, even when only applied to adjacent qubits (lemma 4). Together with its inverse $\mathbf{G}^\dagger$, we can thus use Solovay-Kitaev to approximate the QRM head unitary $\mathbf{R}$ to within precision ϵ. Since we require that the QRM first writes out an instance $l \in \Pi$ on the ring, the resulting circuit $C_{\mathbf{R}}$ has size $|C_{\mathbf{R}}| = \mathrm{O}(\mathrm{poly}\,|l| \log^4(1/\epsilon))$, cf. lemma 5. To match the QRM evolution, we repeatedly apply $C_{\mathbf{R}}$ in a cyclic fashion, as described in fig. 2.

Keeping with tradition, we encode the circuit $C_{\mathbf{R}}$ as a so-called *history state* Hamiltonian. In its simplest form, such a Hamiltonian encodes transitions for each gate $\mathbf{U}_i$ present in $C_{\mathbf{R}} = \mathbf{U}_T \cdots \mathbf{U}_1$. More specifically, on the Hilbert space $\mathbb{C}^{T+1} \otimes (\mathbb{C}^d)^{\otimes n}$ and a basis $\{|j\rangle\}_j$ on $\mathbb{C}^{T+1}$, we define

$$\mathbf{H}_{\mathrm{prop}} := \sum_{t=0}^{T-1} \sum_j (|t\rangle \otimes |j\rangle - |t+1\rangle \otimes \mathbf{U}_t\,|j\rangle)(\text{ h.c. }), \quad (2)$$

where the first component of the Hilbert space stores a clock index taking track of the current step within the computation. One can verify that $\ker(\mathbf{H}_{\mathrm{prop}})$ is spanned by states of the form

$$|\Psi\rangle = \sum_{t=0}^{T} |t\rangle \otimes |\psi_t\rangle := \sum_{t=0}^{T} |t\rangle \otimes \mathbf{U}_t \cdots \mathbf{U}_1\,|\phi\rangle$$

for any initial states $|\phi\rangle \in (\mathbb{C}^d)^{\otimes n}$. We say that the ground state is spanned by states encoding the "history" of the computation, meaning that the state of the computation after t steps—$|\psi_t\rangle$—is entangled with the "time" register $|t\rangle$ (we want to point out, however, that this is a static problem, and the analogy with time steps is purely educational).

A large part of the overhead in terms of local dimension or interaction range present in prior constructions is due to the fact that the terms in eq. (2) are not necessarily local. A common approach to construct a local clock is to subdivide each computational step from $|\psi_t\rangle \longmapsto \mathbf{U}_t\,|\psi_t\rangle = |\psi_{t+1}\rangle$ into multiple intermediate steps

$$|\psi_t\rangle \longmapsto |\psi_{t,1}\rangle \longmapsto \ldots \longmapsto |\psi_{t,s_t}\rangle \longmapsto |\psi_{t+1}\rangle\,,$$

where no quantum gate is applied, but where some internal reordering takes place which allows each transition to act on neighbouring spins only. This allows each gate operation in eq. (2) to be written as a local interaction. However, the problem remains that for each local transition rule, in order to know which gate to apply next, one has to be able to identify the current computational step locally and unambiguously (for an extensive discussion cf. [8, introd.]). Knowing when to apply which transition rule thus requires a potentially large local Hilbert space dimension, or long-range interactions.

Quantum ring machines circumnavigate part of this problem, as only a potentially much smaller circuit $C_{\mathbf{R}}$ has to be applied in a periodic fashion. However, we still need to locally store the current step *within* the circuit $C_{\mathbf{R}}$. In the next section, we explain how we use diagonal Hamiltonian terms to constrain the ground space of our Hamiltonian such that the circuit description for $C_{\mathbf{R}}$ is exposed at the front edge of the cuboid, in a periodically repeating fashion (cf. fig. 3). More precisely, we define a diagonal Hamiltonian $\mathbf{H}_{\mathrm{cl}}$ with spectral gap 1, and a degenerate ground space for which any ground state of $\mathbf{H}_{\mathrm{cl}} + \mathbf{H}_{\mathrm{prop}}$ will then be in a product configuration $|\Phi_{\mathrm{cl}}\rangle \otimes |\Psi\rangle$. Here $|\Phi_{\mathrm{cl}}\rangle$ is a classical product state that takes a configuration as in fig. 3: in particular a string describing $C_{\mathbf{R}}$ is expressed, periodically, on the front edge.

Local terms as in eq. (2) can then be used to access this circuit description *without* any explicit knowledge of the current position within the circuit, which is implicitly given by the location on the cube where the transition rule is applied.

Tiling Construction. A tiling Hamiltonian is a local Hamiltonian on a lattice, where each term is a projector onto the complement of the allowed tiles at a specific lattice location (cf. e.g. [14]). As a simple example, consider just the 2D layer B-type sublattice from fig. 1, and assume that every spin is a qubit. We denote with white the state $|0\rangle$, and with red shading the state $|1\rangle$. Assume the only tiles we want to allow are the four shown in fig. 3 (without rotated variants).

By writing a local term for each tile (where we order the corresponding Hilbert space $(\mathbb{C}^2)^{\otimes 4}$ as a tensor product of the spin on the back, right, front, and left, respectively), we can write a diagonal projector $\mathbf{h} = \mathbb{1} - \sum_{i=1}^{4} \mathbf{h}_i$ on $(\mathbb{C}^2)^{\otimes 4}$ such that the ground space is spanned by quantum states corresponding to the valid tiles; as an example, for the fourth tile, we write

$$\mathbf{h}_4 := |1\rangle\langle 1| \otimes |1\rangle\langle 1| \otimes |0\rangle\langle 0| \otimes |1\rangle\langle 1| \,.$$

We can thus easily define a local Hamiltonian on the layer B-type sublattice which in its zero energy ground state encodes *valid* tiling patterns, where adjacent edges match, if possible; if not, the ground state energy of the Hamiltonian will be at least 1. More specifically, if P indexes all squares with four adjacent spins, then we can write the Hamiltonian as

$$\mathbf{H}_{\text{tiling}} := \sum_{\vec{p} \in P} \left(\mathbb{1}^{\vec{p}} - \sum_{i=1}^{4} \mathbf{h}_i^{\vec{p}} \right) \otimes \mathbb{1} \tag{3}$$

where $\mathbf{h}_i^{\vec{p}} \otimes \mathbb{1}$ acts non-trivially only on the spins sitting on the edges of square $\vec{p}$. For the aforementioned tiles, the resulting pattern is a binary counter, which can be used to translate the depth of a lattice, D, into a binary string representation of D at the front edge (cf. top face in fig. 3).

The same method can equivalently be used to enforce a more complicated tiling pattern in three dimensions, especially when mixing penalty terms with different weights; for an extensive proof that the corresponding Hamiltonian ground space is indeed spanned by the best possible tiling we refer the reader to [14, appdx.].

Hard Instances for the Local Hamiltonian Problem. We will now explain how these two techniques—tiling and history state Hamiltonians—can be combined in order to prove theorem 1. As a first step, we define a tiling pattern to constrain all red layer B spins of the cube—apart from the top and side layers, but including the bottom layer—to a specific symbol which is used nowhere else, and which we denote with ×. All the following terms can then be conditioned on these red spins being either in state ×, or not; this allows us to distinguish between the different faces of the cuboid in a translationally-invariant way and with open boundary conditions. This technique is commonly used in 1D (e.g. [13]), and we extend it to three dimensions.

As explained in the last section, we then define four tiles which self-assemble to a binary counter; this allows us to translates the depth of the cube D to a string representation of D on the top front edge. Using similar tiles on the sides of the lattice, we wind this binary string down and around the cube in an anti-clockwise direction; like that, the string—which is the binary program description of the QRM circuit $C_{\mathbf{R}}$—is expressed periodically on the front edge of the lattice, which we label the *computation edge*, cf. fig. 3.

We further restrict the spins in the green layer A sublattice adjacent to this computation edge to be in a state corresponding to successive pairs of program bits. For example, if the binary program description is p_1, p_2, p_3, p_4, the green spins depend on $(0, p_1), (p_1, p_2), (p_2, p_3), (p_3, p_4)$

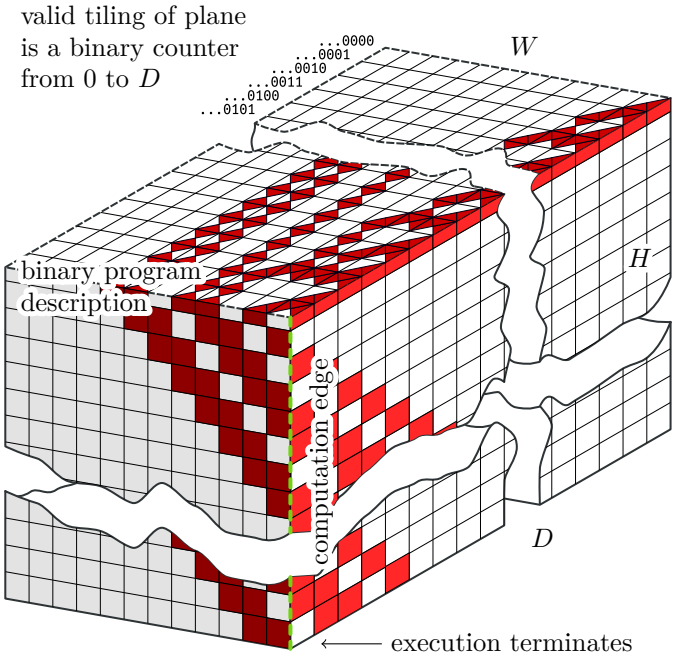


Figure 3: *Structure of the ground state imposed by classical bonus and penalty terms. Shown here is the lattice as in fig. 1; a cut through the top layer of the cuboid shows the layer B red sublattice depicted in fig. 1 (d). Coloured triangles on the top layer denote a spin on the tile edge in configuration $|1\rangle$, a white tile edge stands for configuration $|0\rangle$; the tiles used on the top layer are the following four:*

The same colour coding is used for the squares around the sides, which label the red cF spins on the sides of the unit cells. The dashed green front edge denotes the computation edge, where gates will be applied in the history state construction. Observe how the same binary pattern is repeated periodically along the computation edge.

and $(p_4, 0)$ respectively. A special encoding (cf. table 2) allows us to translate any such binary pair into an operation to perform on the computation edge. All constraints up to this point are diagonal in the computational basis and at most 4-local; we collect all these *static* terms on the cF lattice in the Hamiltonian $\mathbf{H}_{\text{stat}}$.

In order to execute the circuit encoded by the binary string, we will assume that we are working in the ground space of $\mathbf{H}_{\text{stat}}$; any other states necessarily have energy ≥ 1. On the black layer A spins, we partition the Hilbert space $\mathcal{H}_{\text{loc}}$ into $\mathbb{C}^2 \oplus \mathbb{C}^2$; each spin either stores a qubit $|q\rangle$, or it indicates one of two "mover" symbols ◀ and ▶.

We write transition rules for the two arrows, which move them around the cube according to their direction, while staying on the same layer (cf. appendix F.3.1). Any qubit in their path is pushed down to the next layer and cycled one to the right if passed by ▶, or one to the left if passed by ◀. Once an arrow arrives at the computation edge, a transition rule conditioned on the program bit pairs (p_i, p_{i+1}) (accessible through the green spins) performs

the corresponding computational step on the two adjacent qubits. The arrow is then re-set to the next lower level, and the whole procedure repeats. Once the arrow returns to the computational edge and is at the bottom-most layer, there is no further forward transition; the program terminates.

Symbolically, the operations we can perform with this basic set of instructions are the following ones. We have a quantum state of N qubits $|q_1\rangle |q_2\rangle \cdots |q_N\rangle$. In one step, we can either...

1. cycle the qubits clockwise, to $|q_N\rangle |q_1\rangle \cdots |q_{N-1}\rangle$,
2. cycle them anti-clockwise, to $|q_2\rangle \cdots |q_N\rangle |q_1\rangle$,
3. perform a universal two-qubit quantum gate $\mathbf{G}$ on the first two qubits,
4. or perform the inverse of this gate, i.e. $\mathbf{G}^\dagger$.

We prove in lemma 4 that there exists such a gate $\mathbf{G}$ which is universal for quantum computation, even if only applied to adjacent qubits. Analogously to before, we collect all history state terms in the Hamiltonian $\mathbf{H}_{\text{hist}}$.

For us, the lattice instances of interest are the ones where the binary string corresponds to a circuit approximating the head of a $\mathsf{QMA_{EXP}}$ verifier QRM, i.e. $C_{\mathbf{R}}$ (the fact that most program strings do not represent such a QRM is not important). In lemma 5 we perform a careful analysis of the approximation errors, and show that one can indeed choose height, width and depth of the lattice (depth corresponding to the encoded program, width to the ring size, and height to the run time of the verifier) such that the history state corresponds to a witness verification for any instance $l \in \Pi$, where Π can be any promise problem in $\mathsf{QMA_{EXP}}$.

What remains to be done is to penalize invalid history state configurations, such as multiple active symbols, or no active symbol; collect those terms in an operator $\mathbf{P}$, which is the only one which will make use of the scaling freedom given in eq. (2). Finally, an input penalty Π_{in} for the computation ensures that some ancillas are correctly initialized for the computation, and the output penalty Π_{out} raises the lowest energy for NO-instances.

Since our history state has branches (since not all transition rules we write down are completely unambiguous), we have to show that $\mathbf{H}_{\text{prop}}$ defines a so-called unitary labeled graph Laplacian and invoke a recently-proven variant of Kitaev's geometrical lemma for this case, lemma 3. With a rigorous proof in appendix G, we can thus show that the overall 4-local translationally-invariant Hamiltonian

$$\mathbf{H} := \mathbf{H}_{\text{stat}} + \mathbf{H}_{\text{prop}} + \mathbf{P} + \Pi_{\text{in}} + \Pi_{\text{out}}$$

defined on the spin-3/2 cF lattice satisfies the promise gap $\lambda_{\min}(\mathbf{H}) \leq -\Omega(1/\operatorname{poly}|\Lambda|)$ if l is a YES-instance, and $\lambda_{\min}(\mathbf{H}) \geq 0$ otherwise. This finishes the construction, and the claim of theorem 1 follows.

Conclusion

The quest for ever-more physically realistic families of QMA hard local Hamiltonians has arguably led us to increasingly contrived constructions. The increase in complexity necessary when going from non-translationally-invariant constructions to translational invariance is striking [13], and the same holds true for the effort to bring the local dimension back within reasonable range [8]. On the other hand, almost always some fundamental new piece of machinery had to be developed, advancing our knowledge about circuit Hamiltonians: such as allowing branching to happen in the computational path, or using easier-to-implement computational models (Quantum Ring Machines), of independent interest e.g. in the context of adiabatic quantum computation ([15]).

In our case, we combine our construction with Wang tiles, which to our knowledge have not ever been used for this purpose. This "outsourcing" of part of the computation to a classical constraint satisfaction problem saves a significant amount of overhead for the control machinery surrounding the actual quantum verification procedure. Furthermore, the single universal quantum gate could be of independent interest in other applications, as it is reasonable to imagine a physical set-up where gates can only be applied to adjacent qubits in a circuit.

In fact, our 3D construction showcases that the embedded computation need not be highly obscure, and can, in contrast, even be quite elegant, as is evident by the much lower required local dimension and the therefore much smaller number of possible interactions necessary. By moving beyond simple spatial lattices, we can show that such structures *support* the emergence of more complex behaviour, despite the intrinsic symmetry of the crystal lattices we employ. By making use of these novel features, we are able to reduce the local dimension by two orders of magnitude as compared to the best result known to date.

We suggest three concrete open problems.

1. While our cube crystal structure is three-dimensional, we do not exploit its bulk structure beyond making use of its different sides. But there are small universal machines in higher dimensions (e.g. 2D or 3D Turing machines, Turmites, or cellular automata) which might be of use for improving this result further. This also leaves open the question of the required local dimension necessary for any 2D construction.

2. The history state construction we employ still relies on a single moving "head" state. More recent results (cf. [16]) utilize a propagating wave-front-like clock construction in 2D. A more general open question is of course whether there is any other construction, different from Feynman's circuit-to-Hamiltonian one, which would allow one to prove a QMA-hardness result for the local Hamiltonian problem (as studied e.g. in [17]). Including classical computation parts with Wang tiles is one step, but are there other, fundamentally different sets of local interactions even suitable to encode parts of a quantum computation?

3. A bottom-up approach proving a *lower* bound on the local dimension (or locality) of the interactions would be an alternative route to new insights into the LOCAL HAMILTONIAN problem. We want to emphasize that there is not much space left for any optimization:

as mentioned in the introduction, our construction allows each coupling to have $\approx 10^4$ free parameters; by the same benchmark, physically realistic spin lattices found in nature allow somewhere around $(3 \times 3)^2 \approx 80$ different couplings.

Recent results show that e.g. 1D gapped Hamiltonian ground states can be approximated efficiently (i.e. in randomized poly-time, cf. [10]), but since history state constructions have a spectral gap that closes inverse-polynomially with the runtime of the encoded computation, a lower bound on the required local dimension remains open.

Acknowledgements. We are very grateful for discussions with Māris Ozols, who contributed lemma 4. J. B. acknowledges support from the German National Academic Foundation and the EPSRC (grant no. 1600123). S. P. was supported by the EPSRC.

References

[1] Alexei Yu. Kitaev, Alexander Shen, and Mikhail N. Vyalyi. "Classical and quantum computing". In: *Quantum Information*. New York, NY: Springer New York, 2002, pp. 203–217. DOI: 10.1007/978-0-387-36944-0_13.

[2] Julia Kempe, Alexei Yu. Kitaev, and Oded Regev. "The Complexity of the Local Hamiltonian Problem". In: *SIAM Journal on Computing* 35.5 (Jan. 2006), pp. 1070–1097. ISSN: 0097-5397. DOI: 10.1137/S0097539704445226. arXiv: 0406180 [quant-ph].

[3] Roberto I. Oliveira and Barbara M. Terhal. "The complexity of quantum spin systems on a two-dimensional square lattice". In: *Quantum Information & Computation* (Apr. 2005), pp. 1–23. arXiv: 0504050 [quant-ph].

[4] Dorit Aharonov et al. "The power of quantum systems on a line". In: *Communications in Mathematical Physics* 287.1 (May 2009), pp. 41–65. DOI: 10.1007/s00220-008-0710-3. arXiv: 0705.4077.

[5] Sergey Bravyi. "Efficient algorithm for a quantum analogue of 2-SAT". In: Feb. 2011, pp. 33–48. DOI: 10.1090/conm/536/10552. arXiv: 0602108 [quant-ph].

[6] Norbert Schuch. "Complexity of commuting Hamiltonians on a square lattice of qubits". In: (May 2011). arXiv: 1105.2843.

[7] Sean Hallgren, Daniel Nagaj, and Sandeep Narayanaswami. "The Local Hamiltonian problem on a line with eight states is QMA-complete". In: *Quantum Information and Computation* 13.9&10 (Dec. 2013), p. 28. arXiv: 1312.1469.

[8] Johannes Bausch, Toby Cubitt, and Maris Ozols. "The Complexity of Translationally-Invariant Spin Chains with Low Local Dimension". In: *arXiv:1605.01718* (May 2016), p. 52. arXiv: 1605.01718.

[9] Toby Cubitt and Ashley Montanaro. "Complexity classification of local Hamiltonian problems". In: (Nov. 2013), p. 51. arXiv: 1311.3161.

[10] Zeph Landau, Umesh Vazirani, and Thomas Vidick. "A polynomial time algorithm for the ground state of one-dimensional gapped local Hamiltonians". In: *Nature Physics* 11.7 (June 2015), pp. 566–569. ISSN: 1745-2473. DOI: 10.1038/nphys3345. arXiv: 1307.5143.

[11] Daniel Nagaj and Pawel Wocjan. "Hamiltonian quantum cellular automata in one dimension". In: *Physical Review A* 78.3 (Sept. 2008), p. 032311. ISSN: 1050-2947. DOI: 10.1103/PhysRevA.78.032311.

[12] Jianxin Chen et al. "No-go theorem for one-way quantum computing on naturally occurring two-level systems". In: *Physical Review A* 83.5 (May 2011), p. 050301. ISSN: 1050-2947. DOI: 10.1103/PhysRevA.83.050301. arXiv: 1004.3787.

[13] Daniel Gottesman and Sandy Irani. "The Quantum and Classical Complexity of Translationally Invariant Tiling and Hamiltonian Problems". In: *Theory of Computing* 9.1 (May 2013), pp. 31–116. ISSN: 1557-2862. DOI: 10.4086/toc.2013.v009a002. arXiv: 0905.2419.

[14] Johannes Bausch et al. "Size-Driven Quantum Phase Transitions". In: *arXiv:1512.05687* (Dec. 2016), p. 10. arXiv: 1512.05687.

[15] Tzu-Chieh Wei and John C. Liang. "Hamiltonian quantum computer in one dimension". In: *Physical Review A* 92.6 (Dec. 2015), p. 062334. ISSN: 1050-2947. DOI: 10.1103/PhysRevA.92.062334. arXiv: 1512.06775.

[16] Nikolas P. Breuckmann and Barbara M. Terhal. "Space-Time Circuit-to-Hamiltonian Construction and Its Applications". In: (Nov. 2013). DOI: 10.1088/1751-8113/47/19/195304. arXiv: 1311.6101.

[17] Johannes Bausch and Elizabeth Crosson. "Increasing the quantum UNSAT penalty of the circuit-to-Hamiltonian construction". In: (Sept. 2016). arXiv: 1609.08571.

[18] Ethan Bernstein and Umesh Vazirani. "Quantum Complexity Theory". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1411–1473. ISSN: 0097-5397. DOI: 10.1137/S0097539796300921.

[19] John Watrous. "Quantum Computational Complexity". In: *Computational Complexity*. Ed. by Robert A. Meyers. New York, NY: Springer New York, 2012, pp. 2361–2387. ISBN: 978-0-387-75888-6. DOI: 10.1007/978-1-4614-1800-9_147. arXiv: 0804.3401.

[20] Maris Ozols. Private communication. 2016.

[21] Andrew M. Childs et al. "Characterization of universal two-qubit Hamiltonians". In: (Apr. 2010). arXiv: 1004.1645.

[22] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press, 2010, p. 676. ISBN: 9780511976667. DOI: 10.1017/CBO9780511976667.

[23] Matthew J. Patitz. "An introduction to tile-based self-assembly and a survey of recent results". In: *Natural Computing* 13.2 (June 2014), pp. 195–224. ISSN: 1567-7818. DOI: 10.1007/s11047-013-9379-4.

A. Quantum Complexity Classes

In order to rigorously define the complexity classes BQEXP and $\mathsf{QMA}_{\mathsf{EXP}}$, we need to understand the notion of a *uniform circuit family*. Following and referring the reader to [18, 19] for terminology, we give the following definition.

Definition 1 (Uniform family of quantum circuits)**.** *Let $f : \mathbb{N} \to \mathbb{N}$ be a function. A family of quantum circuits $(C_n)_{n\in\mathbb{N}}$ is called f-*uniform *if*

i. each C_n acts on n qubits and has a distinct output qubit,

ii. each C_n requires at most $f(n)$ additional ancillas $|0\rangle$,

iii. each C_n is composed of at most $f(n)$ gates from some universal gate set and

iv. there exists a classical Turing Machine which, on input 1^n produces a description of C_n in fewer than $f(n)$ steps.

A *promise problem* is a pair of disjoint sets $(\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$, corresponding to input strings for YES and NO instances of a set of problem instances $\Pi = \Pi_{\mathsf{YES}} \cup \Pi_{\mathsf{NO}} \subseteq \{0,1\}^*$. We are interested in the quantum generalization of EXPTIME and NEXP—P and NP with exponential runtime.

Definition 2 (BQEXP)**.** *A promise problem $(\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$ is in the complexity class* BQEXP*, bounded-error quantum* exp*-time, if there exists an* exp*-uniform family of quantum circuits $(C_n)_{n\in\mathbb{N}}$ such that*

$$\Pr(C_n(s) = \mathsf{YES}) \geq \frac{2}{3} \quad \text{for } s \in \Pi_{\mathsf{YES}} \quad \text{and} \quad \Pr(C_n(s) = \mathsf{YES}) \leq \frac{1}{3} \quad \text{for } s \in \Pi_{\mathsf{NO}},$$

where $C_n(s)$ denotes the distribution obtained from executing C_n on input $s \in \Pi$ of size $n = |s|$ and measuring the output qubit.

Remark 1. *It is a well-known fact—see [19, prop. 3]—that one can amplify the accept and reject probability of $2/3$ to any $1 - 2^{-p(n)}$ for any fixed polynomial p.*

$\mathsf{QMA}_{\mathsf{EXP}}$ is then the class of promise problems, for which any YES or NO answer can be verified with a BQEXP verifier.

Definition 3 ($\mathsf{QMA}_{\mathsf{EXP}}$)**.** *A promise problem $(\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$ is in $\mathsf{QMA}_{\mathsf{EXP}}$, quantum Merlin-Arthur, if there exists an* exp*-uniform family of quantum circuits, called verifier, each of which with an* exp*-sized witness as input, such that for $l \in \Pi_{\mathsf{YES}}$ there exists a witness such that the verifier accepts with probability at least $2/3$, and for $l \in \Pi_{\mathsf{NO}}$, the verifier accepts with probability at most $1/3$, for any witness.*

The last two conditions on accepting YES and rejecting NO instances is also known as *completeness* and *soundness*, respectively. Probability amplification can be directly translated from their BQEXP counterparts, cf. remark 1.

B. The Local Hamiltonian Problem

We regard Hermitian operators acting on a multipartite Hilbert space $\mathcal{H} = (\mathbb{C}^d)^{\otimes n}$, i.e. on n qudits, each of *local dimension* d. We label subsystems of $\mathcal{H}$ by an ordered tuple $\mathrm{A} \subseteq \{1, \ldots, n\}$. For a k-qudit Hamiltonian $\mathbf{h}$ for some $k \leq n$ and some subset A, we denote with $\mathbf{h}_{\mathrm{A}}$ the operator that acts as $\mathbf{h}$ on all qudits labelled by A, and as identity—$\mathbb{1}$—everywhere else.

Definition 4. *A Hermitian operator $\mathbf{H}$ on Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ is called* k-local *if $\mathbf{H} = \sum_i \mathbf{h}_i$, where we require that there exists a family of subsystems $(\mathrm{A}_i)_i$ of $\mathcal{H}$ such that $|\mathrm{A}_i| \leq k$ $\forall k$, and $\mathbf{h}_i = (\mathbf{h}'_i)_{\mathrm{A}_i}$ $\forall i$.*

If the Hilbert space $\mathcal{H}$ is translationally-invariant—e.g. a lattice $\mathcal{H}^{\otimes\Lambda}$—then we say that a Hamiltonian on this space exhibits translational invariance if it follows the same symmetry, i.e. that the interactions between equivalent lattice sides are identical. This allows us to define the following variant of the local Hamiltonian problem, where we follow the naming convention in [13], i.e. we abbreviate translationally invariant local Hamiltonian as TILH.

Definition 5 ((k,d)-TILH-3D)**.** *Let $\Lambda(L, M, N)$ be a 3D lattice of side lengths L, M and N, all $\leq n$, with not necessarily trivial unit cell (e.g. cF, cI). Let $\mathbf{H} = \sum_{i,j,k} \mathbf{h}_{i,j,k}$ be a 3D translationally-invariant and geometrically k-local Hamiltonian on the lattice qudits $(\mathbb{C}^d)^{\Lambda}$.*

Input. *Specification of the lattice size L, M, N, and the matrix entries of $\mathbf{h}$, up to $\mathrm{O}(\operatorname{poly} n)$ bits of precision.*

Promise. *The operator norm of each local term is bounded, $\|\mathbf{h}\| \leq \operatorname{poly} n$ and either $\lambda_{\min}(\mathbf{H}) \leq \alpha$ or $\lambda_{\min}(\mathbf{H}) \geq \beta$, where $\lambda_{\min}(\mathbf{H})$ denotes the smallest eigenvalue of $\mathbf{H}$ and $\beta - \alpha \leq 1/p(n)$ for some polynomial $p(n)$.*

Output. YES *if $\lambda_{\min}(\mathbf{H}) \leq \alpha$, otherwise* NO*.*

C. Ring Machines

Instead of working with quantum circuits or Turing Machines directly, we will work with a computational model known as *Quantum Ring Machine*.

Definition 6 (Quantum Ring Machine)**.** *A* Quantum Ring Machine—QRM *for short—is a tuple* $(\mathbf{U}, n\,|q_i\rangle\,, \mathcal{H}_f)$ *of a unitary operator acting on a pair of qudits* $(\mathbb{C}^d)^{\otimes 2}$*, and some* $n \in \mathbb{N}$*.* $n \in \mathbb{N}$ *specifies the number of qudits on the ring* $\mathcal{H} := (\mathbb{C}^d)^{\otimes n}$*. Starting out from the initial configuration* $|q_i\rangle \in \mathcal{H}^{\otimes n}$*, we cyclicly apply the unitary* $\mathbf{U}$ *to two adjacent ring sites until the reduced density matrix on one qudit is completely supported in a halting subspace* $\mathcal{H}_f \subsetneq \mathbb{C}^d$*; before that, the overlap of any qudit with* $\mathcal{H}_f$ *is zero.*

QRMs have the distinct advantage of being simple to specify locally—like circuits, but unlike Turing Machines—whilst maintaining a straightforward evolution—in contrast to circuits, which can have a very complex global structure. The similarity between QRMs and TMs is deliberate, as it allows to extend the notions of halting or runtime in a very straightforward manner. A family of QRMs $(\mathbf{U}, n)_{n\in \mathrm{I}}$—labelled by some index set $\mathrm{I} \subset \mathbb{N}$—is called exp-time terminating if it halts on any possible input specified on the tape, and if the number of rounds the unitary $\mathbf{U}$ makes on the tape is upper-bounded by some function $f(n)$ for all $n \in \mathrm{I}$, where $f(n) = \mathrm{O}(\exp n)$. Similar to halting, we specify *accepting* and *rejecting* configurations as special subspaces of $\mathcal{H}$.

If we want to perform a verifier computation with the QRM, we simply leave part of the tape unconstrained as a witness, however much is required by the verification. Central to this work is the following lemma, see [8, cor. 34].

Lemma 1. *Let* $(\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$ *be a* BQEXP *promise problem. Then there exists an* exp*-time terminating family of QRMs* $(M_s)_s$ *such that for* $l \in \Pi_{\mathsf{YES}}$*, there exists a witness* w*, such that the QRM* $M_{|l|}(l, w)$ *accepts with probability at least* $2/3$*. Analogously,* NO *instances are accepted with probability at most* $1/3$*.*

D. Kitaev's History State Constructions with Branching

By embedding a BQEXP-complete QRM into a local Hamiltonian, which allows the execution of a $\mathsf{QMA}_{\mathsf{EXP}}$ verifier circuit, we want to construct a family of $\mathsf{QMA}_{\mathsf{EXP}}$-hard (k, d)-TILH-3D instances. Using i as a multiindex to sum over all 3D lattice sides of Λ, the Hermitian operators that we regard can be written in the form

$$\mathbf{H} = \mathbf{P} + \mathbf{T} = \sum_i \mathbf{p}_i + \sum_i \mathbf{t}_i, \tag{4}$$

where the local terms in $\mathbf{P}$ are diagonal projectors in the computational basis—i.e. classical—and the terms in $\mathbf{T}$ are so-called transition rules. Any transition rule always takes the form

$$\mathbf{t} = \sum_{|e\rangle} (|a\rangle \otimes |e\rangle - |b\rangle \otimes \mathbf{U}\,|e\rangle)(\langle a| \otimes \langle e| - \langle b| \otimes \langle e|\,\mathbf{U}^\dagger) \tag{5}$$

$$\equiv (|a\rangle\langle a| + |b\rangle\langle b|) \otimes \mathbb{1} - |a\rangle\langle b| \otimes \mathbf{U} - |b\rangle\langle a| \otimes \mathbf{U}^\dagger, \tag{6}$$

where $|a\rangle\,, |b\rangle$ are basis vectors in some subspace $\mathcal{H}_c$—which we call classical—and the $|e\rangle$ label a basis of some different—quantum—subspace $\mathcal{H}_q$. $\mathbf{U} \in \mathrm{SU}(\mathcal{H}_q)$ is a unitary operator on this quantum subspace. An operator $\mathbf{T}$ made up of such transition rules can be thought of a generalized Laplacian for a simple graph with unitary edge labels, formally defined as follows.

Definition 7. *A* unitary labeled graph *(ULG) is an undirected graph* $G = (\mathrm{V}, \mathrm{E})$*, a Hilbert space* $\mathcal{H}_v$ *for each graph vertex* $v \in \mathrm{V}$ *and a function* $g : \mathrm{E} \longrightarrow \bigcup_v \mathcal{B}(\mathcal{H}_v)$*, assigning a unitary operator to each edge* $e \in \mathrm{E}$*, where* $g(e) \in \mathcal{B}(\mathcal{H}_a)$ *if* $e = (a, b)$*.*

In particular, the Hilbert spaces attached to two vertices are necessarily isomorphic if the vertices are connected. The associated Hamiltonian is then simply defined as a sum of transition rules of the form eq. (5) for each edge of the graph. We refer the reader to [8, ch. 5] for details and a simple example.

If the product of unitaries along any loop within the graph is the identity operator—where we flip $\mathbf{U}$ to $\mathbf{U}^\dagger$ in case we march against an edge direction—we call the ULG *simple*. A simple ULG has the following important property.

Lemma 2. *The associated Hamiltonian of a simple and connected ULG is unitarily equivalent to copies of the underlying graph's Laplacian* Δ*, i.e. there exists a unitary* $\mathbf{W}$ *such that* $\mathbf{WHW}^\dagger = \Delta \otimes \mathbb{1}_n$*, where* $n = \dim \mathcal{H}_q$*.*

Proof. Cf. [8, lem. 41]. □

Note that this extends to non-connected ULGs in a straightforward manner, as the associated Hamiltonian is block-diagonal in the connected graph components. With this, we can formulate a variant of *Kitaev's geometrical lemma*, which allows us to analyse the spectra of the Hamiltonians we construct.

	1	2	3	4	5	6	7	8	9	10	11
2	3										
3	4	5									
4	6	7	8								
5		9	10	11							
6	12	13	14	15	16						
7		17	18	19	20	21					
8			22	23	24	25	26				
9		27	28	29	30	31	32	33			
10			34	35	36	37	38	39	40		
11				41	42	43	44	45	46	47	
12		48	49		50		51	52	53	54	55
13		56	57	58	59	60	61	62		63	

Table 1: *A Linearly independent set of generators for* $\mathfrak{su}(8)$ *in terms of nested commutators of* $\mathbf{H}_1$ *and* $\mathbf{H}_2$*. For example,* $\mathbf{H}_{42} := \mathrm{i}[\mathbf{H}_{11}, \mathbf{H}_5]$.

Lemma 3. *Take a history state Hamiltonian of the form eq. (4), where* $\mathbf{T}$ *is the associated Hamiltonian of some simple connected ULG with Hilbert space* $\mathcal{H}_q$ *for all vertices* $v \in \mathrm{V}$*. We require that* $\mathbf{P} = \sum_{p\in\mathrm{P}} \Pi_p \otimes \Pi_{p,q}$*, where* Π_p *is a projector on some vertex* $p \in \mathrm{P} \subseteq \mathrm{V}$*, and the* $\Pi_{p,q}$ *are projectors on subspaces of* $\mathcal{H}_q$*. Then* $\lambda_{\min}(\mathbf{H}) = \mu\Omega(1/|\mathrm{V}|^3)$*, where* $\mu = \min\{\lambda_{\min}(\Pi_{p_i,q}\mathbf{U}_{ij}\Pi_{p_j,q}) : p_i, p_j \in \mathrm{P}\}$ *and* $\mathbf{U}_{ij}$ *is the product of unitaries of a path connecting vertices* p_i *and* p_j*.*

Proof. Cf. [8, lem. 44]. □

E. Single Gate Universality

In order to execute the QRM, we have to be able to cyclicly apply the QRM head unitary on a pair of qudits. Since we will be working with qu*b*its in our construction, we embed each such qu*d*it into a list of qu*b*its, and approximate the 2-local qu*d*it unitary using a special 2-local unitary gate $\mathbf{G}$, which can act on any two neighbouring qubits. In order to apply Solovay-Kitaev to the QRM head unitary and approximate it with a $\mathrm{O}(\log(1/\epsilon))$ gate count (as opposed to $\sim 1/\epsilon$), we have to be able to apply both $\mathbf{G}$ and its inverse, $\mathbf{G}^\dagger$; however, in Solovay-Kitaev, the requirement is that those two gates can be applied to *any pair* of qubits, whereas in our construction—as will become clear later—we can only ever apply either gate to neighbouring qubits.

It suffices to prove that $\mathbf{G}$ is universal when applied to adjacent qubits, which is what the following lemma shows.

Lemma 4 (Ozols [20])**.** *Define the 2-qubit unitary* $\mathbf{G} := \exp(\mathrm{i}\mathbf{H})$ *with*

$$\mathbf{H} := \sigma_x \otimes \mathbb{1} + 1 \otimes \sigma_z + \sigma_x \otimes \sigma_x + \sigma_z \otimes \sigma_z = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 0 & -1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

Then the unitaries $\{\mathbf{G}_{k,k+1 \mod l} : k = 0, \ldots, l-1\}$ *generate a dense subset of* $\mathrm{SU}(2^l)$ *for all* $l \geq 3$*, where the subscript denotes where the unitaries act.*

Proof. Since 3-qubit unitaries generate a dense subset of $\mathrm{U}(2^l)$ when applied to adjacent qubits, it suffices to prove the claim for $l = 3$. The proof follows techniques in Lie algebra [21]. Define $\mathbf{H}_1 := \mathbf{H} \otimes \mathbb{1}_2$ and $\mathbf{H}_2 := \mathbb{1} \otimes \mathbf{H}$, and let $\mathcal{L}(\mathbf{H}_1, \mathbf{H}_2)$ be the Lie algebra generated by these two elements. For $j = 3, \ldots, 63$, we set $\mathbf{H}_j := \mathrm{i}[\mathbf{H}_{r_j}, \mathbf{H}_{c_j}]$, where r_j and c_j are the row and column numbers of entry j in Table 1. One can verify—using a computer algebra system—that the matrices $\{\mathbf{H}_1, \ldots, \mathbf{H}_{63}\}$ are linearly independent, and traceless by construction. Since $\dim \mathfrak{su}(8) = 63$, they furthermore span the entire algebra, and the claim follows. □

F. The Cube

F.1. Overview

We work with a face-centered cubic lattice of side lengths $D \times H \times W$, as shown in fig. 3. At each vertex we place a 4-dimensional spin with local Hilbert space $\mathcal{H}_{\mathrm{loc}} = \mathbb{C}^4$, and we want to define a 4-local Hamiltonian on the lattice which embeds the evolution of a $\mathsf{QMA}_{\mathsf{EXP}}$ verifier. Our construction comprises the following three main steps.

Binary Counting. We construct a 2D tileset which lives on the top face of the cuboid, and translates the cuboid depth D into a binary description of D on the top front edge, which is of size $\log_2 D$. This binary string encodes a circuit C according to table 2 and fig. 4.

Shuffling the Program. Using another 2D tileset, we cyclicly shuffle this circuit program around the sides of the cuboid and wind it down diagonally as shown in fig. 3. The front edge—marked in red—is the computation edge and will periodically see the entire binary description of the program.

Performing Gates. On the sides of the cuboid, we superpose a layer of qubits. Labelling the qubits around the top edge of the cube with $|q_1\rangle |q_2\rangle \cdots |q_N\rangle$, we define transition rules which allow us to perform one of the following four operations:

1. cycle the qubits clockwise, to $|q_N\rangle |q_1\rangle \cdots |q_{N-1}\rangle$,
2. cycle them anti-clockwise, to $|q_2\rangle \cdots |q_N\rangle |q_1\rangle$,
3. perform a universal two-qubit quantum gate $\mathbf{G}$ on the first two qubits,
4. or perform the inverse of this gate, i.e. $\mathbf{G}^\dagger$, on the first two qubits.

Once any gate operation is performed, all qubits are swapped with the ones on the next-lower level while cycling them in the direction specified. On the next layer, the same procedure repeats until the execution terminates after H steps (H being the height of the cube). The history state construction for one operation above thus requires $2 \times (D + W)$ steps.

The necessary transition rules are described in detail in appendix F.3.2, and table 2 describes how the binary program description on the computation edge is interpreted as one of the four actions above at each level. Fig. 4 shows how any circuit can be encoded in this way. Observe that due to the winding program description—which is exposed periodically at the front edge—we necessarily apply the same circuit over and over again. In between each appearance of the description of $C_{\mathbf{R}}$ on the computation edge, the string of zeroes does not implement any gates or move the tape in either direction. Naturally, this is precisely the evolution of a Quantum Ring Machine.

For suitable circuits $C_{\mathbf{R}}$, this construction is thus a history state Hamiltonian which encodes an arbitrary QRM.

Remark 2. *If we want to encode a QRM which runs for t applications of the QRM head, we necessarily need $H \geq 2t(D + W)$ for our cube. Furthermore, if the QRM head acts on two qudits of dimension d, the circuit $C_{\mathbf{R}}$ acts on $m = \lceil \log_2 d \rceil$ qubits; we thus require $D + W \equiv 0 \pmod m$.*

For a fixed cube depth D encoding some BQEXP QRM, we need to ensure that we can tune the remaining two free parameters W and H—width and height of the cuboid—to provide enough space and time for the computation to run and terminate, while at the same time keeping the error introduced by approximating the QRM head unitary within bounds. This is captured in the following technical lemma.

Lemma 5. *Take a BQEXP promise problem Π. For any precision $\delta > 0$ and instance $l \in \Pi$, there exist cube parameters $W, H, D = \mathrm{O}(\exp \mathrm{poly}(|l|, \log 1/\delta))$ which allow a verifier ring machine to be executed on the cube for instance l to within precision δ.*

Proof. Let $l \in \Pi$. A BQEXP witness computation for this instance l of size $|l|$ can be performed with a QRM with head unitary $\mathbf{R} \in \mathrm{SU}((\mathbb{C}^d)^{\otimes 2})$ for some d. We require that the QRM head $\mathbf{R}$ contains a description of instance l; this means that d—the size of each of the two qudits that $\mathbf{R}$ acts on—depends on the size of the instance, i.e. $d = \mathrm{O}(\mathrm{poly}\,|l|)$. Denote with t the number of steps the ring machine needs to perform to run the entire verifier computation.

1. In lemma 4 we show that there exists a specific 2-qubit gate $\mathbf{G}$ which is universal for quantum computation, even when only applied to adjacent qubits.
2. Using S-K and a circuit encoding as described in fig. 4 using gates $\mathbf{G}$ and its inverse $\mathbf{G}^\dagger$, approximate the QRM head $\mathbf{R}$ with circuit $C_{\mathbf{R}}$ to some error $\epsilon \leq \delta/t$, where δ is the overall precision which we require for the verifier. Each qudit $\mathbb{C}^d$ of the QRM verifier is encoded in $m = \lceil \log_2 d \rceil$ qubits. The circuit $C_{\mathbf{R}}$ thus acts on $(\mathbb{C}^2)^{\otimes 2m}$, i.e. m qubits. By [22], approximating an n-qubit unitary to within precision ϵ requires $\mathrm{O}(n^2 4^n log^c(n^2 4^n/\epsilon))$ gates (for some $c \leq 4$), if using their gateset; for our purposes it suffices to know that the number of gates required to approximate $\mathbf{R}$ to within precision ϵ scales as $\mathrm{O}(\mathrm{poly}(d) \times \log^c(1/\epsilon))$.
3. The circuit description is thus of length $|C_{\mathbf{R}}| = \mathrm{O}(\mathrm{poly}\,|l| \times \log^c(1/\epsilon))$ and therefore we have to require that the depth of the cube $D = \mathrm{O}(\exp(|C_{\mathbf{R}}|)) = \mathrm{O}(\exp(\mathrm{poly}\,|l| \times \log^c(1/\epsilon)))$.
4. The front sidelength W is increased...
   - to make the ring $r = W + D$ large enough for the computation, if it is not already, and

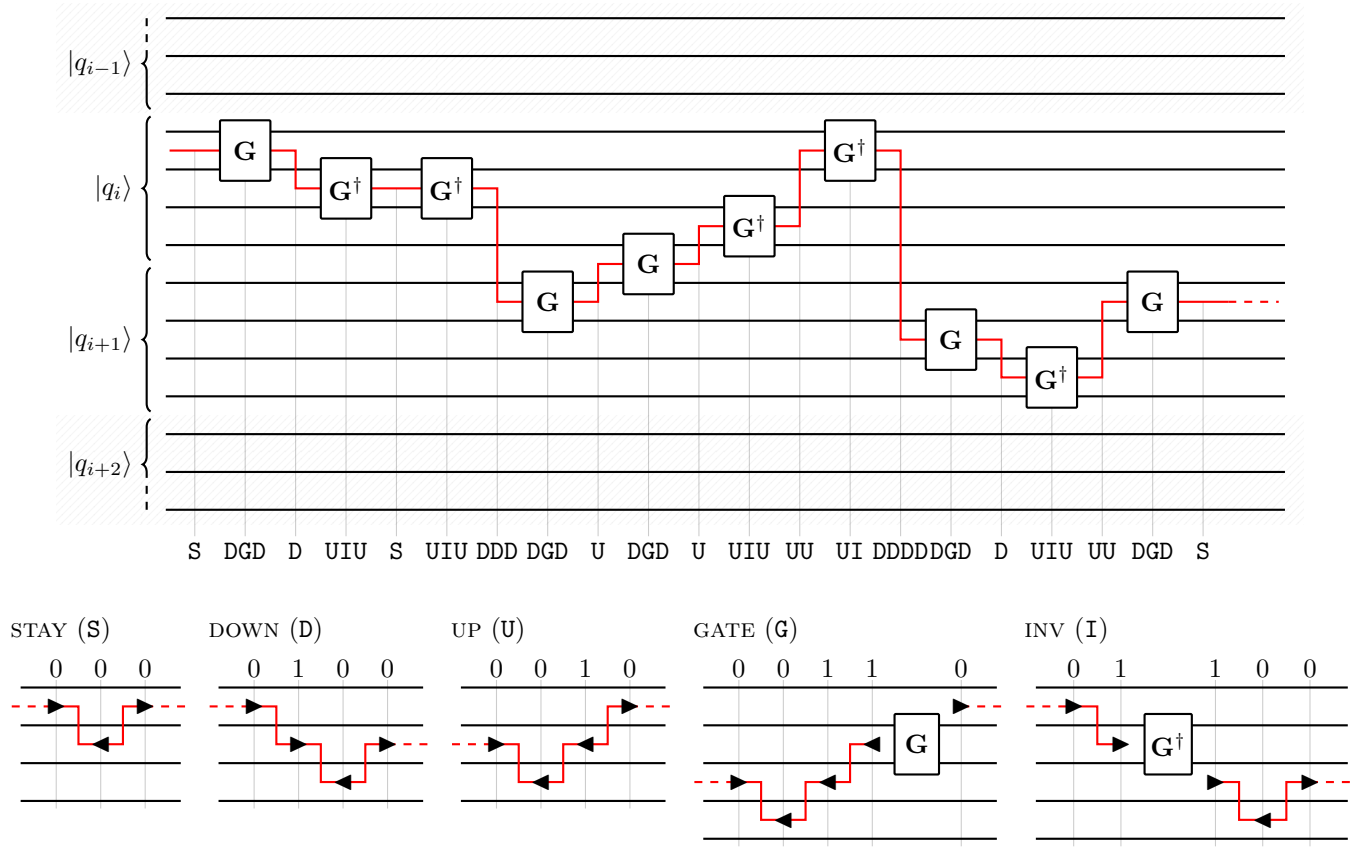


Figure 4: *Execution order of an arbitrary circuit approximated using the universal gate* $\mathbf{G}$ *and its inverse* $\mathbf{G}^\dagger$*. Each elementary operation start and end in a configuration* ▶*, where the last program bit is a 0—like this, each circuit can be constructed by a simple combination of these elementary operations, with a constant overhead. Observe that both gate application and inverse gate application do not end on the same line, which means that if we want to apply* $\mathbf{G}$ *at the current position, we have to execute* `DGD`*, and similarly* `UIU` *for* $\mathbf{G}^\dagger$*. The specific quantum gate* $\mathbf{G}$ *that we use is proven to be universal in lemma 4.*

- to make the ring size an integer multiple of $m = \lceil \log_2 d \rceil$.

5. Set $H = 2t(W + D)$.

With $\epsilon \leq \delta/t$ and $t = \mathrm{O}(\exp \mathrm{poly}\,|l|)$, we further have $\log^c 1/\epsilon \leq \log^c(t/\delta) = \mathrm{O}(\mathrm{poly}(|l|, \log 1/\delta))$, and the claim of the lemma follows. □

Remark 3. *If we require cube parameters of* $\mathrm{O}(\exp \mathrm{poly}\,|l|)$*, we can demand a computation accuracy of at most* $\delta = \Omega(1/\exp \mathrm{poly}\,|l|)$.

Proof. If we demand the two scaling parameters in lemma 5 to be equal, we have

$$\begin{aligned} & \exp \log^4 (1/\delta) = \mathrm{O}(\exp \mathrm{poly}\,|l|) \\ \Leftrightarrow \quad & \log^4 (1/\delta) = \mathrm{O}(\mathrm{poly}\,|l|) \\ \Leftrightarrow \quad & \log (1/\delta) = \mathrm{O}(\mathrm{poly}\,|l|) \\ \Leftrightarrow \quad & \delta = \Omega(1/\exp \mathrm{poly}\,|l|). \end{aligned}$$

□

F.2. Static Lattice Constraints

F.2.1. Lattice Structure

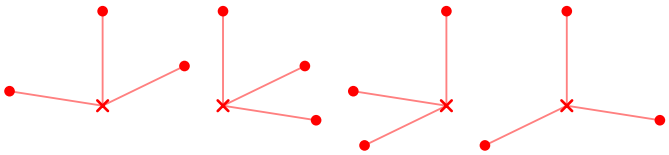
We will work with a face-centered cubic lattice of 4-dimensional qudits. All interactions will be at most 4-local and translationally-invariant. The system will have open boundary conditions; in particular, we do not cut off interactions at the boundary or introduce boundary constraints of any kind. For the sake of clarity, when writing out constraints in the following, we will usually ignore parts of the sublattice, implicitly assuming that any interaction term is extended trivially everywhere else. When refering to layer A and if not explicitly mentioned, we mean the black sublattice, and layer B will be the red sublattice with side-centered vertices.

Any "static" constraint—i.e. the terms in the following four subsections—will be translated into local Hamiltonian terms diagonal in the computational basis; see appendix F.2.6 for details.

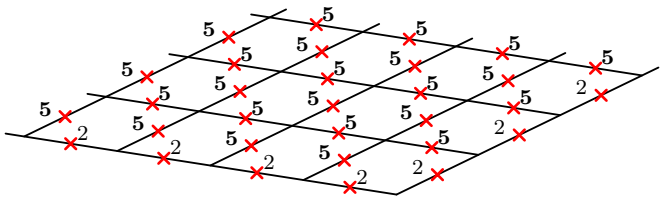
F.2.2. Constraining the Lattice Bulk

Denoting with × a special symbol in the red sublattice, we want to constrain the lattice to be in this state in the bulk, and in its complement on the topmost red face, as well as the outermost side faces. We first give a bonus of 1 to spins in the B sublattice in configuration

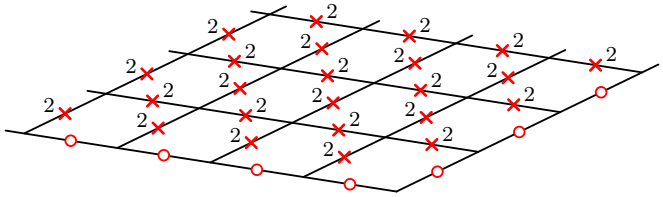
All red layers but the top one will then be in state ×. We then give a bonus of 1 to all of the following configurations:



This leaves the top layer unchanged. Summarizing the bonus terms so far, all other B layers, as seen from the top, are then in the configuration

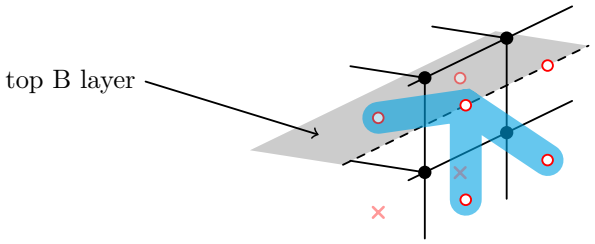


We then give a global 1-local penalty to × with strength -3. The top B layer will thus be in the complement of × (which we denote with ○), while all the other B layers look like



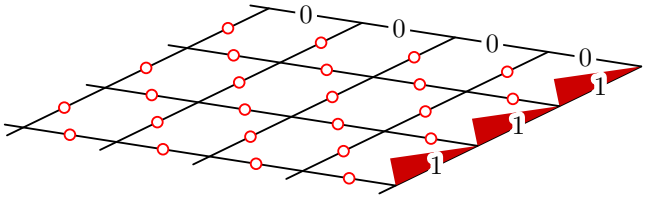
F.2.3. Binary Counter

The top layer of type B will carry a binary counter tiling, which translates the side length D into a binary representation on the top front edge. In order to achieve this, we need to initialize the top back edge of the cube to all 0, and the top right edge to all 1. Since we do not want to use distinct interactions on the outside layers, but have open boundary conditions, we have to find a configuration in the pre-constrained cube which only occurs on the top right and back edge, respectively. The following configuration is such an example:



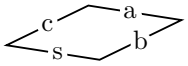
(7)

Since only the top and outer layers have red spins in configuration ○, this four-local interaction allows us to pick out the top right boundary of the top layer, and to constrain it to state 1. A similar interaction allows constraining the top back layer to 0. The top B layer then looks like



Since this is the only B layer with spins in state ○, we can use the following tiles from [23] to get the desired binary counting layer.

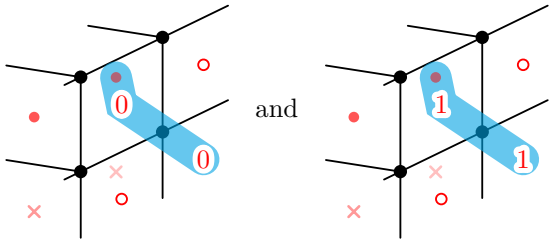
It is straightforward to verify that the general tile



obeys the rules $c =$ carry of $a + b$ and sum $s = a \oplus_2 b$.

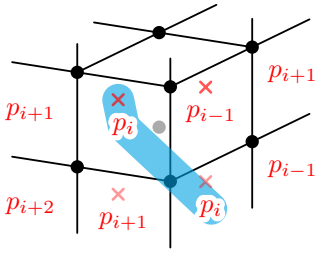
F.2.4. Winding Program Diagonally

We use an interaction similar to eq. (7) to shuffle the program around the cube in a cyclic fashion, as depicted in fig. 3:



Observe that, by including the red qudit one layer in, this interaction does indeed only apply to the front right face; similar interactions on the other three faces achieve the desired program copying around the cube sides. Additionally, by conditioning on if this inner qudit is either × or ○, we can apply a different rule at the top layer. In particular at the top layer of the front right face we want to flip the bit when copying down so that there are 1's on the top layer but 0's on the layer below - see fig. 3.

On the corners, we use a similar shape of interaction, i.e.



and similarly for all other corners.

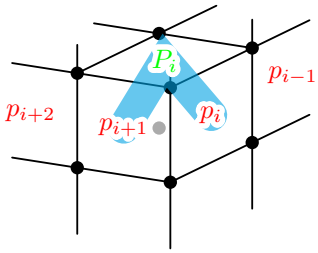
Note that in appendix F.3.2, we will need to temporarily replace red program bits with a special symbol ! indicating that the application of a gate is happening in the next step, so we exclude this case from the constraints in this section (i.e. we allow *either* p_i and p_i, or p_i and ! to appear around the computational corner, and similarly for the diagonal face constraints bordering the computation edge).

As none of the dynamic transition rules below ever changes the number of head symbols (of which ! is one), we can rule out the cases where there is more than one ! or other head symbol present at any one time—we analyse these branching cases in detail in appendix F.4.

F.2.5. Constraining layer A qudits

We label the states of the green face-centred qubits of the layer A type with the alphabet $\{A, B, C, 0\}$. For all such green lattice qubits we apply a bonus of strength $1/2$ to configuration 0, so that this state is preferred.

In order to access two sequential program bits p_i and p_{i+1} with a single three-local interaction on the computation edge, we add a strength 1 interaction which constrains the front column of the layer A green sublattice to a state $P_i \in \{A, B, C\}$ depending on the two neighbouring computation bits, i.e.



Note that this interaction will have no effect anywhere else in the lattice, as at least one of the two red program bits will be ×.

The rule which governs what state P_i is constrained to will depend on the tuple p_i and p_{i+1}, and is derived from table 2. The idea is that $P_i = f(p_i, p_{i+1})$ will signify what is to happen at the computation edge. Looking at table 2, we see that at each stage we either:

A. **A**pply a gate (either $\mathbf{G}$ or its inverse $\mathbf{G}^\dagger$, depending on where the arrow is coming from),

B. go **B**ackwards (i.e. change the direction of the arrow),

C. or **C**ontinue in the same direction.

Given the encoding of table 2, we therefore take $P_i = f(p_i, p_{i+1})$ for a function f given by

$$f(p_i, p_{i+1}) = \begin{cases} B & \text{if } p_{i+1} = 0, \\ C & \text{if } p_i = 0 \text{ and } p_{i+1} = 1, \text{ and} \\ A & \text{if } p_i = p_{i+1} = 1. \end{cases}$$

Due to the aforementioned $1/2$ bonus which applies at all green spins, the remainder of layer A is in configuration 0.

F.2.6. Summary of static constraints

As explained in the main text under "Tiling Construction", we take all static constraints listed so far and translate them to diagonal and local projectors $\mathbf{h}_i$. This allows us to write a 4-local, translationally-invariant classical Hamiltonian $\mathbf{H}_{\text{stat}} = \sum_{\vec{x}} \sum_{\mathbf{h}_i} \mathbf{h}_i^{\vec{x}}$ (i.e. product and diagonal in the computational basis of each spin) with a ground space spanned by states with the following properties.

1. Any black vertex spin in layer A is unconstrained.

2. The red layer B spins will be in a state as depicted in fig. 3, i.e. on the top cuboid face, they represent a binary counter translating the depth D of the cuboid into a binary description of D on the top front edge. This binary string $s = p_1 \dots p_T$ is wound down diagonally around the cube, which expresses s periodically on the front computation edge. *Only* the spins adjacent to this edge are also allowed in a configuration !. In the bulk of the cube all the way to the bottom-most layer, the red spins are in state ×.

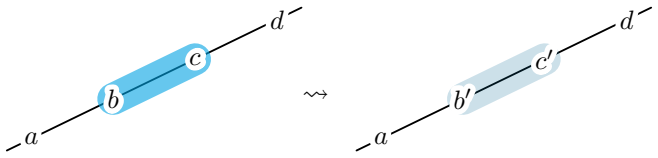
3. The green layer A is in configuration 0 everywhere but on the front edge; there, the spins there are in a configuration depending on the two adjacent program bits p_i and p_{i+1}, as outlined above.

This Hamiltonian $\mathbf{H}_{\text{stat}}$ is gapped with a size-independent constant gap, and we can rescale the interactions so far and shift the overall energy to assume that this ground space as detailed above has energy zero, and any other configuration has energy lower-bounded by 1.

In the next sections, we will explain the history state construction, which—within this ground space of $\mathbf{H}_{\text{stat}}$—will represent a valid QRM evolution for the circuit represented by the binary string s.

F.3. Dynamic Constraints on Computational Layer

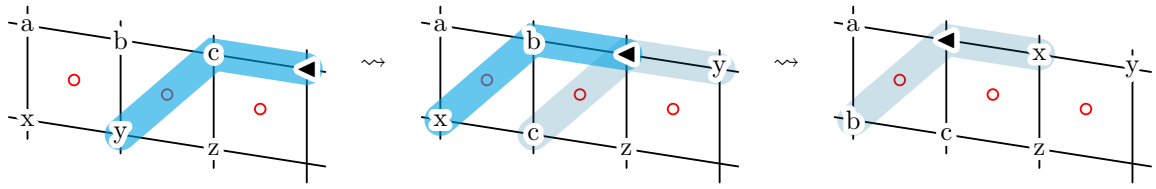
The "dynamic" history state transition rules will be translated in a similar fashion to terms as in eqs. (4) and (5). We always depict a transition rule as connected by a squiggly arrow $\rightsquigarrow$; the notation is self-explanatory: the brighter blue shading indicates the original state, whereas the dull blue shading indicates the target configuration. To give an example, a transition



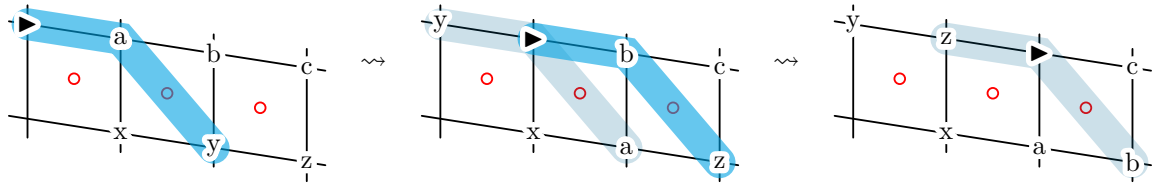
would be translated into a two-local term $\mathbf{h} = |bc\rangle\langle bc| + |b'c'\rangle\langle b'c'| - |b'c'\rangle\langle bc| - |bc\rangle\langle b'c'|$, and correspondingly with an extra quantum register if b or c were labelling vertices that carry a qubit (i.e. the black layer A sublattice vertices).

F.3.1. Moving Qubits

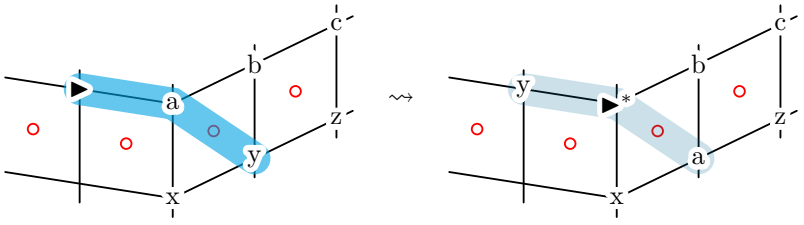
The black sublattice (A layers) comprises the alphabet $\{0, 1, \blacktriangleright, \blacktriangleleft\}$, where we treat the $0, 1$-subspace as a qubit, i.e. $\mathbb{C}^2$. The right and left arrows are markers to indicate where to move qubits to. As an example on the front face, we have a left moving sequence



and analogously the right moving sequence



To move qubits around a corner, we use an interaction of the form



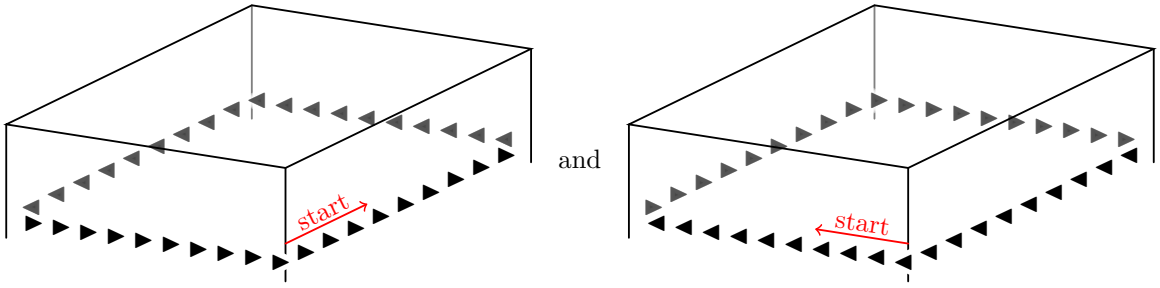
at the back, left and right corners (different rules as described in appendix F.3.2 are used for the front edge) and similarly for going around the corner in the opposite direction.

A few remarks: first note that all the transitions defined so far are unique, i.e. given the cube bulk constrained to × as done in appendix F.2.2, and for every configuration with only one arrow symbol (the other cases we will penalize as a last step), there exists precisely one forward and one backwards transition. Another important point is how to modify the arrows when going around the circumference of the cube once (marked with a $\blacktriangleright^*$ in the last transition rule); at the moment, if we left the arrow type unchanged for every corner, we would not be able to shuffle around the qubits in a circle; on the back face, we would be doing the opposite shuffling operation. Therefore, we change the arrow

tape coming from	program $x_{n-1}x_n$	operation on qubits	tape going to
▶	00	$\mathbb{1}$	◀
▶	01	$\mathbb{1}$	▶
▶	10	$\mathbb{1}$	◀
▶	11	$\mathbf{G}^\dagger$	▶
◀	00	$\mathbb{1}$	▶
◀	01	$\mathbb{1}$	◀
◀	10	$\mathbb{1}$	▶
◀	11	$\mathbf{G}$	◀

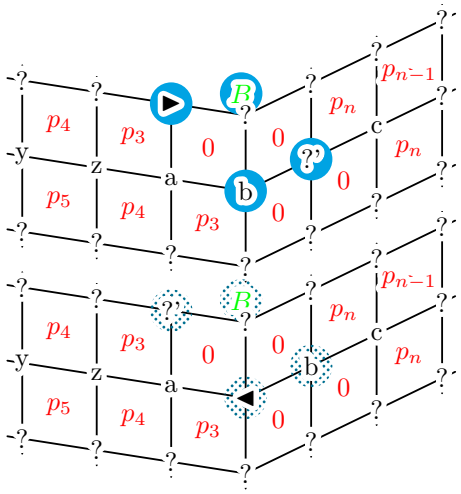
Table 2: *Program encoding. The arrow symbols ▶ and ◀ indicate in which direction the ring is moving. Relative to the tape, the current head is thus moving in the opposite direction. With this encoding, any circuit can be executed with the available operations, cf. fig. 4.*

type according to the following scheme:



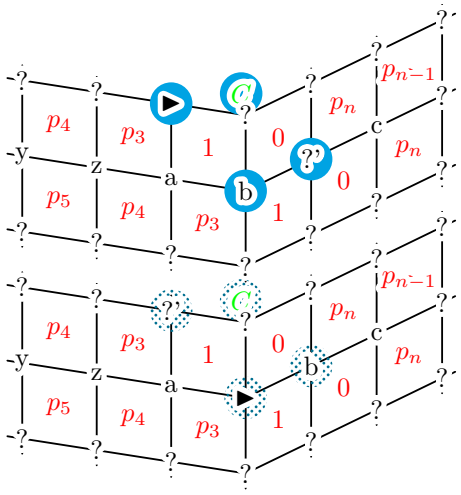
F.3.2. Computation

In order to execute any circuit as in fig. 4, we have eight elementary operations available, all of which are listed in table 2. It is easy to see that there exists a symmetry between the right- and left-moving arrow; we will thus explain the right-moving arrows (including the application of gate **G**) in detail and leave the reverse direction as an exercise to the reader.

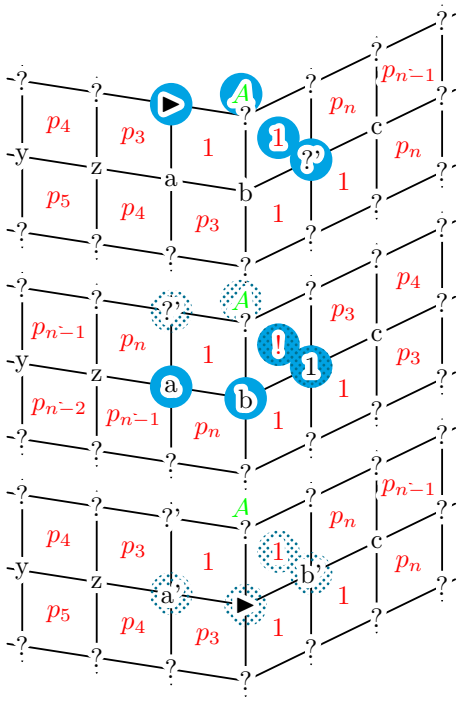


Consider $p_1p_2 = 00$ or 10, so that $f(p_1, p_2) = B$. The transition is conditioned to only happen if the green qubit in the layer A sublattice is in the B state.

Move ?' up, b to the right and flip the arrow. This corresponds to simply reverting the direction as in table 2.



Consider $p_1p_2 = 01$ so that $f(p_1, p_2) = C$. We perform the same action as above, but keep the arrow direction.



Consider $p_1p_2 = 11$ so that $f(p_1, p_2) = A$. We want to execute a gate, which requires one intermediate step.

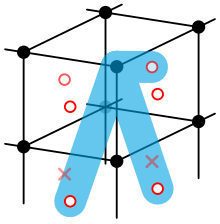
We place the computation marker on the right hand side of the computation edge. This signals that the next step is to perform a gate $\mathbf{G}$ on a and b.

Here $|a'\rangle |b'\rangle := \mathbf{G} |a\rangle |b\rangle$. The program is restored and the arrow left in the right moving configuration, as required by table 2.

We now move the arrow once around the tape and then arrive at the computational corner from the other side. Observe—as mentioned—that the encoding in table 2 is mirror-symmetric, so by reversing all the rules above one can implement the same rules—while applying $\mathbf{G}^{-1}$ instead of $\mathbf{G}$ when $P_i = A$ for an arrow incoming from the right.

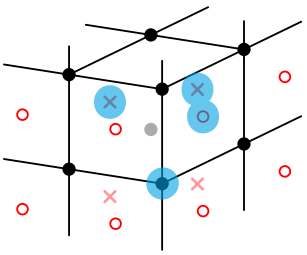
F.3.3. Computational Input and Output Constraints

Since the instance is specified within the QRM head, it suffices to provide the computation with a single ancilla $|0\rangle$ as input; in case we need more ancillas than available on the front edge, we can augment our verifier as in [8, fig. 4]. Due to the configuration of the red layer B sublattice, it is straightforward to find a local configuration which only ever appears on a top right corner; more specifically, we utilize the constraint interaction

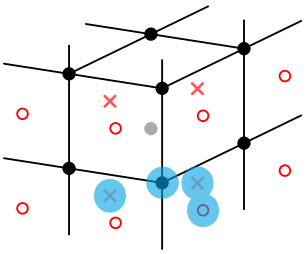


to enforce that the black symbol is either in an arrow configuration, or 0, respectively. The rest of the tape is left unconstrained.

Since there is nothing special about the bottom-most layers A and B, we need to use a pair of interactions to enforce the last black qubit to an accepting state. This can be readily achieved using



constraining the black qubit to state $|0\rangle$, and



giving a bonus to the complement configuration.

Everywhere but on the bottom-most layer, the two penalties precisely cancel; however, on the last layer, only the projection onto $|0\rangle$ survives, which thus acts as output penalty once the computation is terminated.

F.3.4. Multiple Heads Penalty

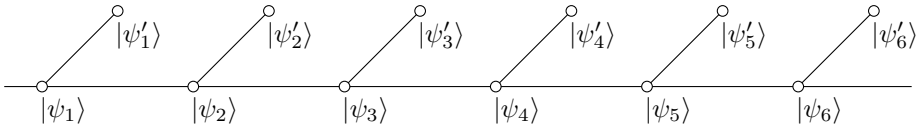
Since we only want to allow precisely one head on the computational layer, we will penalize any configuration where two heads are next to each other. This finishes our construction.

F.4. Valid History State Branching

In this section, we want to analyse all transition rules and show that the parts where they are ambiguous do not break the evolution of the computation. First note that all constraints in appendix F.2 are static, i.e. there are no possibilities for any ambiguities in the configuration. We will call configurations that obey all those static constraints and have precisely one head symbol on the computational layer—i.e. exactly one of ▶, ◀ or !—*valid* configurations.

We will go through each dynamic penalty in appendix F.3 separately.

1. In appendix F.3.1, the transition rules for the faces are unambiguous, since they depend on the red symbol to be in a configuration ○.
2. The rules for moving around a corner, however, *can* happen on a face: in this case, the arrow symbol is moved one layer into the bulk. Observe though that none of the movement transitions can apply to the arrow when it is inside of the bulk (apart from moving it back out with a reverse transition), so the computation branches, but the leg does not proceed: we obtain an evolution of the form



   where all the primed states are redundant, but at most enlarge the overall evolution by a factor of 2.
3. In appendix F.3.2, the computation transitions are unambiguous; observe in particular that there is no transition rule that simply copies the arrow around the computation edge (by construction, see appendix F.3.1).
4. Finally, the input and output constraints are static again.

This allows us to formulate the following two branching lemmas.

Lemma 6. *Any valid history state for the given transition rules is of size* $\mathrm{O}(\mathrm{poly}(W, D, H))$*, where* W*,* D *and* H *are the cuboid's width, height and depth.*

Proof. Follows by construction; the head can perform at most $\mathrm{O}(H \times (W + D))$ unique transitions. □

Lemma 7. *In case there is more than one head symbol (i.e. !,* ▶ *or* ◀*) present, the minimal valid evolution splits up into* poly*-sized slices, each of which carries at least one penalty from two directly adjacent heads.*

Proof. The argument is the same as in [8]. One can keep all but one of the head symbols fixed; the one left free to move is necessarily meeting another head symbol within poly many steps. □

G. QMA-Hardness Proof of Main Theorem

In this section, we provide a rigorous proof of theorem 1. Using statical constraints and dynamic rules as in eqs. (4) and (5), we translate the transition rules defined in appendix F into a Hamiltonian $\mathbf{H}_{\text{prop}}$, which is geometrically 4-local by construction.

We want to point out that the Hilbert space structure of this lattice Hamiltonian $\mathbf{H}_{\text{prop}}$ is not a product space between clock and computation space $\mathcal{H}_{\text{clock}} \otimes \mathcal{H}_{\text{comp}}$, which would result in a ground state of the standard history state form $\sum_t |t\rangle |\psi_t\rangle$. The reason for this is that depending on which sub-lattice a spin sits on, its local Hilbert space $\mathcal{H}_{\text{loc}} = \mathbb{C}^4$ decomposes differently. The red and green spins can be regarded as being completely in the clock space, as all transition rules which act on them are completely classical, i.e. they never move any of the red and green spins out of a computational basis state. The black spins, however, decomposes into a direct sum $\mathcal{H}_{\text{clock}} \oplus \mathbb{C}^2$, the latter space carrying a qubit, and the clock part being reserved for the two arrow symbols ◀ and ▶, which are part of the clock.

In order to analyse the spectrum, we note that there exists an isometric transformation between our Hamiltonian and Hilbert space, and one which respects the product space structure, which in particular will allow us to regard the Hamiltonian as a ULG Laplacian and apply lemma 3. Let us be precise at this point, and use the recently-developed Quantum Thue System terminology defined in [8, sec. 6]. With this new machinery, we can state the following lemma.

Lemma 8. *The transition rules in appendix F define a Quantum Thue System, and the induced ULG is simple.*

Proof. Verifying that the rules define a Quantum Thue System is straightforward by a simple re-ordering of the spins. Simplicity of the corresponding unitary labelled graph follows from lemma 6; we refer the interested reader to [8, def. 51, lem. 52 and 53]. □

Without further ado, we now proceed to the proof of theorem 1, which we re-state here in a rigorous, but concise fashion.

Theorem 2. $(4,4)$-TILH-3D *is* $\mathsf{QMA}_{\mathsf{EXP}}$*-complete.*

Proof. Containment in $\mathsf{QMA}_{\mathsf{EXP}}$ is straightforward, cf. [8]. To show that the Hamiltonian instances of the cube construction define a $\mathsf{QMA}_{\mathsf{EXP}}$-hard family, we will employ techniques proven there which should simplify the analysis.

Let $\Pi = (\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$ be a $\mathsf{QMA}_{\mathsf{EXP}}$ promise problem, as in definition 3. By lemma 5, we know that we can pick a constant error threshold $\delta > 0$ such that for any instance $l \in \Pi$ there exists a cube which allows a verifier circuit for this instance to be executed on the sides. Since we will require probability amplification later on in the proof (remark 1), we set $\delta = f(|l|)$ for some function f to be specified later, and also assume that the original verifier's acceptance probability is $\epsilon_l \leq f(|l|)$.

We translate all static and dynamic penalties into a Hamiltonian as explained in eq. (4), and denote the corresponding Hamiltonian operator with

$$\mathbf{H} = \mathbf{P} + \mathbf{H}_{\text{prop}} = \mathbf{P}_{\text{in}} + \mathbf{P}_{\text{out}} + \mathbf{P}_{\text{static}} + \mathbf{P}_{\text{heads}} + \mathbf{H}_{\text{prop}},$$

where $\mathbf{P}_{\text{static}}$ comprises all static constraints for the cube (cube structure, binary counter and winding of program), $\mathbf{P}_{\text{heads}}$ penalizes any two head symbols next to each other, and such that $\mathbf{P}_{\text{in/out}}$ represent the input and output penalties, respectively.

Soundness. We first regard the case when $l \in \Pi_{\mathsf{YES}}$. Denote with $|\Psi_l\rangle$ the valid history state, i.e. the unique uniform superposition ground state of $\mathbf{H}_{\text{prop}}$ started out in a valid initial configuration with a single left-moving head in the top left row, and such that no initial or static penalty is violated. Then

$$\begin{aligned}
\langle\Psi_l|\,\mathbf{H}\,|\Psi_l\rangle &= \langle\Psi_l|\,\mathbf{P}_{\text{in}}\,|\Psi_l\rangle && = 0 \quad \text{(because } |\Psi_l\rangle \text{ satisfies all input constraints)} \\
&+ \langle\Psi_l|\,\mathbf{P}_{\text{out}}\,|\Psi_l\rangle \\
&+ \langle\Psi_l|\,\mathbf{P}_{\text{static}}\,|\Psi_l\rangle && = 0 \quad (|\Psi_l\rangle \text{ is valid history state}) \\
&+ \langle\Psi_l|\,\mathbf{P}_{\text{heads}}\,|\Psi_l\rangle && = 0 \quad (|\Psi_l\rangle \text{ has one active head symbol}) \\
&+ \langle\Psi_l|\,\mathbf{H}_{\text{prop}}\,|\Psi_t\rangle && = 0 \quad \text{(since } |\Psi_l\rangle \text{ ground state of } \mathbf{H}_{\text{prop}}).
\end{aligned}$$

What remains to be analysed is the output penalty $\langle\Psi_l|\,\mathbf{P}_{\text{out}}\,|\Psi_l\rangle$. If we write $|\Psi_l\rangle = \frac{1}{\sqrt{T}} \sum_{t\in T} |t\rangle |\psi_t\rangle$ where T is the normalization constant for the history state (i.e. the number of unique vertices in the ULG evolution represented by $|\Psi_l\rangle$), which we know by lemma 6 to be $T = \mathrm{O}(\mathrm{poly}(W, D, H))$—i.e. the number of computational steps taken, including branching, cannot be larger than a polynomial in the cube width, depth and height. Then

$$\begin{aligned}
\langle\Psi_l|\,\mathbf{P}_{\text{out}}\,|\Psi_l\rangle &= \frac{1}{T}\left(\sum_{t,t'\in T} \langle t|\,\langle\psi_t|\,|T\rangle\langle T|\otimes\Pi_{\text{out}}\,|t'\rangle\,|\psi_{t'}\rangle\right) \\
&= \frac{1}{T}\,\langle\psi_T|\,\Pi_{\text{out}}\,|\psi_T\rangle = \frac{1}{T}\mathbb{P}(\text{circuit rejects}) \leq \frac{1}{T}(\epsilon_l + \delta) = \frac{2f(|l|)}{T}.
\end{aligned}$$

Completeness. If $l \notin \Pi_{\mathsf{YES}}$, we have to show that for any $|\psi\rangle$, $\langle\psi| \mathbf{H} |\psi\rangle$ is bounded away from the YES-case by a $1/\operatorname{poly}$ gap. If any static constraint is violated, we can immediately bound $\mathbf{H} \geq 1$. So we can assume that the state $|\psi\rangle$ is in a valid configuration.

Note that the number of head symbols ▶, ◀ or ! is always preserved for any transition rule. This means that $\mathbf{H}_{\text{prop}}$—and therefore also $\mathbf{H}$—is block-diagonal in the static cube configuration and the number of head symbols on the computational layer, we can regard each case separately.

1. In case of multiple head symbols we observe that each head necessarily sweeps the entire surface of the cube. Mark an arbitrary head symbol, and define $\mathbf{H}'_{\text{prop}}$ to be $\mathbf{H}_{\text{prop}}$ with any transitions for the other heads removed. Any such transition rule as in eq. (5) is positive semi-definite, which necessarily means $\mathbf{H}_{\text{prop}} \geq \mathbf{H}'_{\text{prop}}$ (spectrum wise, by which we mean $\mathbf{H}_{\text{prop}} - \mathbf{H}'_{\text{prop}}$ is psd itself). This new operator $\mathbf{H}'_{\text{prop}}$ might be non-local, but we only need it to lower-bound the spectrum of $\mathbf{H}_{\text{prop}}$.

   The marked head symbol will then encounter another head in at most $\operatorname{poly}(W, H, D)$ many steps (it cannot take longer than visiting the entire surface of the cuboid, cf. lemma 7). At that point, it will pick up a penalty. Utilizing our variant of Kitaev's lemma (lemma 3), we conclude

   $$\mathbf{H} = \mathbf{P}_{\text{in/out}} + \mathbf{P}_{\text{static}} + \mathbf{P}_{\text{heads}} + \mathbf{H}_{\text{prop}} \geq \mathbf{P}_{\text{heads}} + \mathbf{H}'_{\text{prop}} \geq \Omega(1/\operatorname{poly}(W, H, D)).$$

   We thus need to set f to a function which allows a polynomial separation (in the system size) between YES and NO instance; by remark 3, this is always possible.

2. The same argument lets us bound $\mathbf{H} \geq \mathbf{P}_{\text{in/out}} + \mathbf{H}_{\text{prop}} = \Omega(1/\operatorname{poly}(W, H, D))$ in case of a single head valid history state since l is a NO-instance.

3. What remains to be analysed is the zero head case. There are two standard approaches: we can either increase the number of symbols on the computational layer, such that on one side of a head—i.e. behind it in direction of the computation—we take one kind of symbols, and on the other side we take the other set; constructions like this can be constrained by a regular expression (without repetition of symbols, cf. [13, lem. 5.2]) and thus penalized with local terms.

   Since our benchmark tries to reduce the local dimension of the system, we instead add a bonus term $\mathbf{B}$ to the Hamiltonian $\mathbf{H}$, and such that $\mathbf{B} |\psi\rangle = -g(|l|) \times h |\psi\rangle$ where h is the number of head symbols for any basis state $|\psi\rangle$ of $\mathbf{H}$, and g is a function chosen such that there is again a $1/\operatorname{poly}$ separation between the zero head state and the ground state for YES-instances, but such that multiple head configurations stay bounded away from 0. It is clear that $\mathbf{B}$ can be implemented by a 1-local term of the form $-g(|l|) \, |\text{head}\rangle\langle\text{head}|$.

   To be more precise and to determine how quickly g has to grow, assume that the construction up to now satisfies $\lambda_{\min} \leq 1/A$ for the YES case, and $\lambda_{\min} \geq 1/B$ in the NO case (excluding zero heads), where $B = \mathrm{O}(\operatorname{poly}(W, H, D))$, and $A \geq 4B \times W \times H \times D$. Choose $g = 2/A$; since there can be at most $W \times H \times D$ heads on the cuboid's faces, we obtain the bounds

   $$\lambda_{\min} \begin{cases} \leq 1/A - 2/A = -1/A & \text{if } l \in \Pi_{\mathsf{YES}} \\ \geq 1/B - 4(W \times H \times D)/A \geq 1/B - 1/B \geq 0 & \text{otherwise, and for at least 1 head.} \end{cases}$$

   The zero head case can then easily be lower-bounded by $\mathbf{H} \geq 0$.

We have thus shown a promise gap of $1/\operatorname{poly}$ in the system size: for $l \in \Pi_{\mathsf{YES}}$, $\mathbf{H} \leq -\Omega(1/\operatorname{poly}(W, H, D))$, and $\mathbf{H} \geq 0$ otherwise. The claim of theorem 1 follows. □